

APPLICATION FOR UNITED STATES LETTERS PATENT
FOR
PROGRAMMABLE CLOCK DELAY CIRCUIT

INVENTOR(S): **DARREN SLAWECKI**

PREPARED BY:

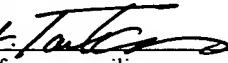
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD, 7TH FLOOR
LOS ANGELES, CALIFORNIA 90025
(206) 292-8600

Express Mail Certificate of Mailing

"Express Mail" mailing label number: EV320117562US
Date of Deposit: March 31, 2004

I hereby certify that this paper or fee is being deposited with the
United States Postal Service "Express Mail Post Office to Addressee"
service under 37 CFR 1.10 on the date indicated above and is
addressed to the Mail Stop Patent Application, Commissioner for
Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Yuko Tanaka
(Typed or printed name of person mailing paper or fee)


(Signature of person mailing paper or fee)

March 31, 2004
(Date signed)

PROGRAMMABLE CLOCK DELAY CIRCUIT

TECHNICAL FIELD

[0001] This disclosure relates generally to delay circuits, and in particular but not exclusively, relates to a programmable clock delay circuits.

5

BACKGROUND INFORMATION

[0002] Within most integrated circuits ("ICs") there is usually one data path that requires more time to propagate valid data than all other data paths. The data path that requires the longest propagation time before it may be sampled or is resolved is

10 known as the critical path of the IC. A circuit path may be slow due, for example, to a greater number of device delays within the critical path or a greater signal travel distance.

[0003] The maximum speed at which the IC may operate is limited by the critical path of the IC. The reason for this is that the critical path presents the longest 15 delay path and the clock rate cannot be increased beyond the point at which the clock cycle time is equal to the propagation delay of signals traveling along the critical path.

[0004] Since the maximum clock speed of an IC is limited by its critical path, locating the critical path (LCP) is an important design task. Once the critical path has been identified, the design may be optimized to reduce the time it takes a signal to 20 propagate along the critical path. LCP and design optimizations may be repetitive tasks. Each time the design is optimized to reduce the delay length of a critical path, a new critical path may arise. Large scale IC design is complicated by the millions of possible

critical paths. LCP becomes the task of locating the proverbial needle in a haystack. As such, sophisticated design and testing tools are required.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Non-limiting and non-exhaustive embodiments of the present invention are described with reference to the following figures, wherein like reference numerals refer to like parts throughout the various views unless otherwise specified.

5 [0006] FIG. 1 is a block diagram illustrating a clock distribution network for distributing a reference clock signal to a logic cluster.

[0007] FIG. 2 is a timing diagram illustrating how delaying a reference clock signal along a critical path of an integrated circuit can be used to increase the reference clock speed.

10 [0008] FIG. 3 is a circuit diagram illustrating a clock delay circuit, in accordance with an embodiment of the present invention.

[0009] FIG. 4 is a timing diagram illustrating variable rising edge and falling edge delays of a clock delay circuit, in accordance with an embodiment of the present invention.

15 [0010] FIG. 5 is a circuit diagram illustrating an inverting clock delay circuit, in accordance with an embodiment of the present invention.

[0011] FIG. 6 is a timing diagram illustrating variable rising edge and falling edge delays of an inverting clock delay circuit, in accordance with an embodiment of the present invention.

20 [0012] FIG. 7 illustrates an integrated circuit including clock delay circuits to selectively delay a reference clock by variable amount throughout the integrated circuit, in accordance with an embodiment of the present invention.

[0013] FIG. 8 is a timing diagram illustrating a rising edge delay of a reference clock using a non-inverting clock delay circuit and a falling edge delay of the reference clock using an inverting clock delay circuit, in accordance with an embodiment of the present invention.

5 [0014] FIG. 9 is a flow chart illustrating a process to determine delay settings of clock delay circuits within an integrated circuit, in accordance with an embodiment of the present invention.

[0015] FIG. 10 is a block diagram illustrating a demonstrative processing system for implementing embodiments of the present invention.

DETAILED DESCRIPTION

[0016] Embodiments of a system and apparatus for implementing a programmable delay circuit are described herein. In the following description numerous specific details are set forth to provide a thorough understanding of embodiments of the 5 invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

10 [0017] Reference throughout this specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrases “in one embodiment” or “in an embodiment” in various places throughout this specification are not necessarily all 15 referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0018] Throughout this specification, several terms of art are used. These terms are to take on their ordinary meaning in the art from which they come, unless specifically defined herein or the context of their use would clearly suggest otherwise.

20 Use of the phrases “logic low” or “logic 0” may be used interchangeably to represent one logic state of binary logic while a “logic high” or “logic 1” may represent the other state. Further, the logic described herein may include a third logic state known as a “high impedance state”.

[0019] FIG. 1 is a block diagram illustrating a synchronous circuit 100 including clock delay circuits for timing a logic cluster, in accordance with an embodiment of the present invention. Synchronous circuit 100 includes a clock distribution network 105, a delay circuit 110, flip-flops FF1 and FF2, and logic cluster 115.

[0020] Clock distribution network 105 delivers a reference clock signal 120 to flip-flop FF1 and delay circuit 110. Clock distribution network 105 may include a number of branching signal paths that are routed throughout synchronous circuit 100. Clock distribution network 105 may include a number of repeaters (not illustrated) to restore reference clock signal 120 and maintain an acceptable slope and skew throughout and delay buffers (not illustrated) to match clock propagation delays to each of flip-flops FF1 and FF2. A clock generator 125 generates reference clock signal 120. Clock generator 125 typically is external to synchronous circuit 100 and may include a crystal resonator, such as quartz, or other known clock generating circuits. Logic cluster 115 may include combination logic and/or sequential logic having finite delays.

[0021] In the illustrated embodiment, flip-flop FF1 is directly clocked by reference clock 120. Delay circuit 110 is configured to receive reference clock signal 120 and output a delayed clock signal FF2CLK to clock flip-flop FF2. In synchronous designs, such as synchronous circuit 100, events occur on clock edges, either the rising edge or the falling edge. Flip-flops FF1 and FF2 are illustrated as rising edge flip-flops, though falling edge flip-flops may also be implemented. Flip-flops FF1 and FF2 hold their outputs FF1OUT and FF2OUT between rising edges of their clock signals. Upon each rising edge of reference clock signal 120, flip-flop FF1 latches its input FF1IN to

its output FF1OUT and holds FF1OUT until at least the next rising edge of reference clock signal 120. Similarly, flip-flop FF2 latches its input FF2IN to its output FF2OUT in response to each rising edge of FF2CLK. Thus, FF1OUT must propagate through logic cluster 115 and resolve as FF2IN within one period of reference clock signal 120 in 5 order to latch to FF2OUT in a timely manner.

[0022] FIG. 2 illustrates a timing diagram 200 of the signals illustrated in FIG.

1. FIG. 2 illustrates how delay circuit 110 introduces a delay Δ_A into FF2CLK for timing flip-flop FF2. When reference clock signal 120 rises at 201, flip-flop FF1 latches input FF1IN through to output FF1OUT after a propagation delay inherent to flip-flop 10 FF1, as illustrated by arrow 205. Subsequently, a change in FF1OUT is propagated through logic cluster 115 and resolves at 210 as input FF2IN to flipflop FF2, as illustrated by arrow 215. As can be seen, FF2IN does not resolve until after the next rising edge 220 of reference clock signal 120. Without delay circuit 110 delaying reference clock signal 120 by delay Δ_A , flip-flop FF2 would latch a stale value of FF2IN 15 through to FF2OUT. However, because delay circuit 110 outputs FF2CLK with delay Δ_A relative to reference clock signal 120, FF2IN is resolved prior to rising edge 225 of FF2CLK. Therefore, the current value of FF2IN is latched through to FF2OUT.

[0023] If logic cluster 115 represents the critical path of synchronous circuit 100, then the propagation delay from FF1OUT to FF2IN (plus the clock to out delay of 20 flip-flop FF1 and the setup time of FF2IN) corresponds to the shortest period of reference clock signal 120 which may drive synchronous circuit 100. Delaying FF2CLK by delay Δ_A , provides the critical path with an additional time equal to delay Δ_A to resolve. The effect of this is that reference clock signal 120 may be increased in

frequency. However, padding time to the critical path with delay circuit 110 is done at the expense of the next logic path through which FF2OUT must propagate. As such, inserting delay Δ_A into FF2CLK relative to reference clock signal 120 is a sort of “robbing Peter to pay Paul” activity. However, the effect of this time borrowing from 5 one propagation path to the next can result in substantially higher global clock frequencies (e.g., reference clock signal 120) for an integrated circuit (“IC”), such as synchronous circuit 100, provided there is sufficient margin for logic coupled to the output side of flip-flop FF2.

[0024] FIG. 3 is a circuit schematic of a clock delay circuit 300, in accordance 10 with an embodiment of the present invention. In one embodiment, clock delay circuit 300 can selectively insert one of four incremental clock delays via programmable delay settings. Furthermore, in one embodiment, clock delay circuit 300 can independently delay a rising edge or a falling edge of a reference clock signal. It should be appreciated 15 that although clock delay circuit 300 is described in connection with delaying clock signals, that various other types of signals may be selectively delayed with clock delay circuit 300.

[0025] The illustrated embodiment of clock delay circuit 300 includes a clock enable circuit 305, a falling edge delay circuit 310, and a rising edge delay circuit 315. The illustrated embodiment of clock enable circuit 305 includes a clock input 320 for 20 receiving reference clock signal 120 (hereinafter REF CLK 120), an enable input 325 to receive an enable signal, and an output 330 to output a delayed clock signal. The illustrated embodiment of falling edge delay circuit 310 includes two falling delay inputs 80 FD0 and FD1 for selecting one of four falling delays to apply to the falling edge of REF

CLK 120. The illustrated embodiment of rising edge delay circuit 315 includes two rising delay inputs RD0 and RD1 for selecting one of four rising delays to apply to the rising edge of REF CLK 120.

[0026] The components of clock enable circuit 305 are interconnected as follows. Clock input 320 and enable input 325 are coupled to the inputs of a NAND gate L1. The output of NAND gate L1 is coupled to a node 335. Node 335 is coupled to a pull up path 340, a pull down path 345, and an input of an inverter L2. The output of inverter L2 is coupled to output 330 for outputting the delayed clock signal. Pull up path 340 includes a P-type metal oxide semiconductor (“PMOS”) transistor T1 having a drain coupled to node 335 and a source coupled to falling edge delay circuit 310. Pull down path 345 includes two N-type MOS (“NMOS”) transistors T2 and T3 coupled in series between node 335 and rising edge delay circuit 315. The drain of transistor T2 is coupled to node 335 and the source of transistor T3 is coupled to falling edge delay circuit 315. Clock input 320 is further coupled to the gates of transistor T1 and T3 to turn transistor T1 on and transistor T3 off or transistor T1 off and transistor T3 on. Enable input 325 is further coupled to the gate of transistor T2. As can be seen from FIG. 3, when enable input 325 is logically low, transistor T2 is turned off and the output of NAND gate L1 will rise. Thus, when enable input 325 is a logic low or ‘0’, the value of node 335 is a logic high or ‘1’. Having node 335 default to a high logic value enables quicker response for clock delay circuit 300, since NMOS transistors are more efficient than PMOS transistors and can pull node 335 down faster than PMOS transistor can pull node 335 up. Further, having node 335 rise when enable input 325 is logic low allows use of smaller P-type transistors thereby saving valuable IC real estate.

[0027] The components of falling edge delay circuit 310 are interconnected as follows. Falling edge delay circuit 310 includes a NAND gate L3, an inverter L4, a NOR gate L5, an inverter L6, and PMOS transistors T4-T7. Logic L3-L6 acts as a decoder of delay settings applied to inputs FD0 and FD1 to selectively turn on and off transistors T4-T7. Transistors T4-T7 are arranged into three parallel pull up paths 350 each coupled between a source voltage VCC and pull up path 340 of clock enable circuit 305. Logic L3-L6 along with inputs FD0 and FD1 are coupled to the gates of transistors T4-T7 to selectively turn on each of pull up paths 350. Increasing the number of pull up paths 350 conducting decreases the overall pull up resistance causing node 335 to rise quickly with less total fall delay through clock delay circuit 300. Correspondingly, decreasing the number of pull up paths 350 conducting increases the overall pull up resistance causing node 335 to rise slowly with more delay. It should be appreciated that the particular combinations of logic L3-L6 may be varied using more or less logic gates to obtain the same decoding results within the spirit of the present invention. Further, it should be appreciated that falling edge delay circuit 310 could be designed having more or less pull up paths 350 with corresponding decoder logic to support more or less falling delay inputs.

[0028] The components of rising edge delay circuit 315 are interconnected as follows. Rising edge delay circuit 315 includes a NAND gate L7, inverter L8, NOR gate L9, and NMOS transistors T8-T11. Logic L7-L9 acts as a decoder of delay settings applied to inputs RD0 and RD1 to selectively turn on and off transistors T8-T11. Transistors T8-T11 are arranged into three parallel pull down paths 355 each coupled between ground (or other low reference voltage) and pull down path 345 of clock enable

circuit 305. Logic L7-L9 along with inputs RD0 and RD1 are coupled to the gates of transistors T8-T11 to selectively turn on each of pull down paths 355. Increasing the number of pull down paths 355 conducting to ground decreases the overall pull down resistance causing node 335 to fall quickly with less total rise delay through clock delay

5 circuit 300. Similarly, decreasing the number of pull down paths 355 conducting increases the overall pull down resistance causing node 335 to fall slowly with more delay. It should be appreciated that the particular combinations of logic L7-L9 may be varied using more or less logic gates to obtain the same decoding results within the spirit of the present invention. Further, it should be appreciated that rising edge delay circuit 10 315 could be designed having more or less pull down paths 355 with corresponding decoder logic to support more or less rising delay inputs.

[0029] FIG. 4 illustrates a timing diagram 400 depicting variable rising edge and falling edge delays inserted by clock delay circuit 300, in accordance with an

embodiment of the present invention. Timing diagram 400 includes a graphical

15 representation of REF CLK 120 input into clock input 320 and a delayed clock signal 405 (hereinafter delayed CLK 405) generated at output 330. As illustrated, rising edges 410 of REF CLK 120 may be selectively delayed by one of four rising delays.

Similarly, falling edges 420 of REF CLK 120 may be selectively delayed by one of four falling delays.

20 [0030] In one embodiment, rising edges 415 of delayed CLK 405 may be

delayed in linear increments of $n \cdot \Delta_1$, where $n = 0, 1, 2$, or 3 and Δ_1 is a finite time

delay. Thus, when $[RD0, RD1] = [0, 0]$, $n = 0$, then rising edges 410 are delayed by a minimal amount τ_1 , which is equal to the time for REF CLK 120 to propagate through

clock enable circuit 305 with all of pull down paths 355 conducting. When $[RD0, RD1] = [0, 1]$, $n = 1$, then rising edges 410 are delayed by τ_1 plus one Δ_1 delay, and two of pull down paths 355 are conducting. When $[RD0, RD1] = [1, 0]$, $n = 2$, then rising edges 410 are delayed by τ_1 plus two Δ_1 delays, and one of pull down paths 355 is conducting.

5 When $[RD0, RD1] = [1, 1]$, $n = 3$, then rising edges 410 are delayed by τ_1 plus three Δ_1 delays, and none of pull down paths 355 are conducting. By selecting the sizes of transistors T8-T11 the increments between each rising delay may be linear or even nonlinear. In one embodiment, clock delay circuit 300 may be designed such that one Δ_1 delay is equal to 7ps.

10 [0031] In one embodiment, falling edges 425 of delayed CLK 405 may be delayed in linear increments of $m \cdot \Delta_2$, where $m = 0, 1, 2$, or 3 and Δ_2 is a finite time delay. When $[FD0, FD1] = [0, 0]$, $m = 0$, then falling edges 420 are delayed by the minimal amount τ_2 , which is equal to the time for REF CLK 120 to propagate through clock enable circuit 305 with all of pull down paths 350 conducting. When $[FD0, FD1] = [0, 1]$, $m = 1$, then falling edges 410 are delayed by τ_2 plus one Δ_2 delay, and two of pull up paths 350 are conducting. When $[FD0, FD1] = [1, 0]$, $m = 2$, then falling edges 410 are delayed by τ_2 plus two Δ_2 delays, and one of pull up paths 350 is conducting. When $[FD0, FD1] = [1, 1]$, $m = 3$, then falling edges 410 are delayed by τ_2 plus three Δ_2 delays, and none of pull up paths 350 are conducting.

20 [0032] As can be seen from FIGs. 3 and 4, clock delay circuit 300 can be programmed with different delay settings applied to each of falling delay inputs FD0 and FD1 and rising delay inputs RD0 and RD1. By selecting the sizes of transistors T4-T7 and T8-T11, the increments between each rising delay and each falling delay may be

linear or nonlinear. Further is should be appreciated that the rising delays may be independently adjusted or selected from the falling delays. Adjusting the delays applied to rising edges 415 and falling edges 425 of delayed CLK 405 does not change the frequency of delayed CLK 405 from that of REF CLK 120. However, the duty cycle of 5 delayed CLK 405 is altered when the rising edge and/or the falling edge delays are applied.

[0033] FIG. 5 is circuit schematic of an inverting clock delay circuit 500, in accordance with an embodiment of the present invention. Inverting clock delay circuit 500 operates in a is similar to clock delay circuit 300, except clock input 320 is logically 10 inverted and inputs FD0, FD1 and RD0, RD1 are swapped. Like components are labeled with like references.

[0034] Inverting clock delay circuit 500 includes an inverting clock enable circuit 505, a rising edge delay circuit 510, and a falling edge delay circuit 515. Inverting clock enable circuit 505 differs from clock enable circuit 305 by the insertion 15 of an inverter L10 between clock input 320 and NAND gate L1. Rising edge delay circuit 510 is similar to falling edge delay circuit 310, with the exception that the inputs RD0 and RD1 select rising delays, as opposed to falling delays. Falling edge delay circuit 515 is similar to rising edge delay circuit 315, with the exception that the inputs FD1 and FD0 select falling delays, as opposed to rising delays. Thus, the delay setting 20 inputs RD0, RD1 and FD0, FD1 are reversed between clock delay circuit 300 and inverting clock delay circuit 500.

[0035] FIG. 6 illustrates a timing diagram 600 depicting variable rising edge and falling edge delays inserted by inverting clock delay circuit 500, in accordance with

an embodiment of the present invention. As can be seen from FIG. 6, rising edges 410 of REF CLK 120 are translated by inverting clock delay circuit 500 to falling edges 605 of delayed CLK 610. Falling edges 420 of REF CLK 120 are translated by inverting clock delay circuit 500 to rising edges 615 of delayed CLK 610.

5 [0036] Falling edges 605 may be selectively delayed according to delay settings applied to RD0 and RD1 of rising edge delay circuit 510. In one embodiment, falling edges 605 may be delayed by one of four incremental delays. In one embodiment, the incremental delays are linearly separated with increments of $n \cdot \Delta_3$, where $n = 0, 1, 2$, or 3 and Δ_3 is a finite time delay. When $[RD0, RD1] = [0, 0]$, $n = 0$, 10 then the falling edges of delayed clock signal 610 are delayed by the minimal amount τ_3 , which is equal to the time for REF CLK 120 to propagate through inverting clock enable circuit 505 with all of the pull up paths of rising edge delay circuit 510 conducting.

15 [0037] Rising edges 615 may be selectively delayed according to delay settings applied to FD0 and FD1 of falling edge delay circuit 515. In one embodiment, rising edges 615 may be delayed by one of four incremental delays. In one embodiment, the incremental delays are linearly separated with increments of $n \cdot \Delta_4$, where $n = 0, 1, 2$, or 3 and Δ_4 is a finite time delay. When $[FD0, FD1] = [0, 0]$, $n = 0$, then the rising edges of delayed clock signal 610 are delayed by the minimal amount τ_4 , which is equal to the time for REF CLK 120 to propagate through inverting clock enable circuit 505 with all 20 of pull down paths of falling edge delay circuit 515 conducting. It should be appreciated that delaying the falling edges 605 or rising edges 615 of delayed clock signal 610 does not cause the frequency of delayed CLK 610 to differ from the frequency of REF CLK 120, rather merely selectively delays its falling and/or rising edges therefrom. Further, it

should be noted that the delays inserted into falling edges 605 are independent of the delays inserted into rising edges 615.

[0038] FIG. 7 illustrates an integrated circuit (“IC”) 700 including delay circuits to selectively delay REF CLK 120 by variable amounts throughout IC 700, in accordance with an embodiment of the present invention. The illustrated embodiment of IC 700 includes delay circuits 705A-D (collectively 705), flip-flops 710A-D (collectively 710), and logic clusters 715A-D (collectively 715).

[0039] Delay circuits 705A-C clock rising edge flip-flops 710A-C and therefore may correspond to embodiments of clock delay circuit 300. Delay circuit 10 705D clocks a falling edge flip-flop 710D and therefore may correspond to embodiments of inverting clock delay circuit 500. One of ordinary skill in the art having the benefit of the instant disclosure will appreciate that other configuration combinations are also possible. For example, a falling edge flip-flop (e.g., flip-flop 710D) may be clocked off clock delay circuit 300, or conversely, a rising edge flip-flop (e.g., flip-flop 710A) may 15 be clocked off inverting clock delay circuit 500.

[0040] Each of delay circuits 705 include a set of inputs 720, which correspond to [RD0, RD1] and [FD0, FD1]. When enable input 325 is asserted, flip-flops 710 each store data received from one of logic clusters 715 coupled to an input for one clock cycle and latch the data to an output for delivery to a next stage of logic clusters 715. 20 Although clock delay circuit 300 and inverting clock delay circuit 500 both are illustrated with enable inputs 325, it should be appreciated that alternative embodiments of the present invention (e.g., clock delay circuit 300 and inverting clock delay circuit

500) need not include enable inputs. Rather, in these alternative embodiments, clock enable circuit 305 and inverting clock enable circuit 505 are always enabled.

[0041] Each of flip-flops 710 are clocked by a delayed version of REF CLK 120. In one embodiment, each of delay circuits 705 may be configured to delay its 5 corresponding flip-flop 710 by a similar amount. In one embodiment, individual delay settings may be applied to each of delay circuits 705 to delay their corresponding flip-flop 710 by individually selected amounts. In yet another embodiment, IC 700 is divided into domains 730. In this alternative embodiment, the inputs 720 of each of delay circuits 705 residing within the same domain 730 are coupled together, such that 10 the same delay settings are applied to all delay circuits 705 within a single one of domains 730. Grouping delay circuits 705 into domains 730 may be convenient for very large scale integrated circuits (“VLSIs”), which may include 18,000 delay circuits or more. Determining and applying individual delay settings to 18,000 delay circuits may be an unreasonably difficult design task requiring considerable circuit real estate devoted 15 to routing conductor traces for inputs 720. For example, 18,000 delay circuits may be grouped into approximately 120 domains 730. It should be appreciated that embodiments of the present invention may include any number of delay circuits 705 grouped into any number of domains 730 for clocking flip-flops 710. It should further be appreciated that embodiments of the present invention may further be used to delay 20 other types of latches and/or sampling circuits than just flip-flops 710.

[0042] FIG. 8 illustrates a timing diagram 800, in accordance with an embodiment of the present invention. Timing diagram 800 shows how rising edges of a

delayed CLK 405 output by delay circuits 705A-C and a falling edges of delayed CLK 610 output by delay circuit 705D relate to a single rising edge 410 of REF CLK 120.

[0043] Delay circuits 705A-C delay REF CLK 120 without inverting. Rising edge 410 of REF CLK 120 results in a delayed rising edge 415 of delayed CLK 405.

5 Rising edge 415 is delayed by $\tau_1 + n \cdot \Delta_1$ (where n=0, 1, 2, 3). Delay circuit 705D is configured to invert and delay REF CLK 120. Falling edge 605 is delayed by $\tau_3 + n \cdot \Delta_3$ (where n=0, 1, 2, 3). Typically, inverting clock delay circuits 500 will have a larger minimum delay τ_3 due to the extra delay added by inverter L10. However, transistors T8-T11 of falling edge delay circuit 515 and transistors T2 and T3 of inverting clock 10 enable circuit 505 (and T4-T7 of rising edge delay circuit 510 and transistor T1 of inverting clock enable circuit 505) may be designed to compensate for this extra delay. By swapping FD0, FD1 and RD0, RD1 between clock delay circuits 300 and 500, the same clock edge is impacted for both inverting and non-inverting delay circuits 705.

15 Without swapping FD0, FD1 and RD0, RD1, a change in any delay settings applied to inputs 720 could result in a race path. For example, if a logic path begins with a non-inverting element (e.g., flip-flop 710C) and ends with an inverting element (e.g., flip-flop 710D), then changing delay settings applied to both of their inputs 720 will not cause frequency variations or race paths to appear.

[0044] FIG. 9 is a flow chart illustrating a process 900 to determine delay 20 settings to apply to delay circuits 705, in accordance with an embodiment of the present invention. In a process block 905, REF CLK 120 having an initial frequency is applied to IC 700. The initial frequency applied may be a frequency just beyond a fail point frequency of IC 700. In a process block 910, the delay settings applied to each of

domains 730 are adjusted or “tweaked” to locate the one or more domains 730 that no longer cause IC 700 to fail at the initial failing frequency due to the adjustments applied to inputs 720. Individually tweaking each domain 730 to determine which domain is on the verge of failure can expedite locating the critical path. In one embodiment, the 5 adjustments applied to inputs 720 may be applied in an ad hoc manner using educated guesses based on knowledge of the design of IC 700 and where the critical path of IC 700 is likely to reside. Alternatively, a systematic approach to tweaking each domain 730 may be taken.

[0045] Once the critical path has been located, the delay settings applied to 10 delay circuits surrounding the critical path may be adjusted to provide extra time for the critical path (process block 915). Subsequently, the frequency of CLK REF 120 may be increased to leverage the additional time padded to either end of the critical path. Then, process 900 loops back to process block 910 where the delay settings applied to domains 730 are once again tweaked to locate the critical path of IC 700. If a new critical path is 15 located then the delay settings may again be adjusted surround this new critical path (process block 915) and REF CLK 120 again increased to leverage the new settings (process block 920). Process 900 loops around many times, as indicated by arrow 927 until adjusting the delay settings applied to inputs 720 of delay circuits 705 can no longer increase the frequency of REF CLK 120. At this point, it is determined in a 20 decision block 925 that the maximum frequency of REF CLK 120 has been attained. Once the maximum frequency is attained, the delay settings are fused (e.g., permanently set) into IC 700 as production tuning (process block 930). Tweaking the delay settings

of delay circuits 705 can result in substantial increases in the frequency of REF CLK 120 (e.g., as much as 550 MHz or more).

[0046] Thus, embodiments of clock delay circuit 300 and inverting clock delay circuit 500 (i.e., delay circuits 705) enable independent rising edge and falling edge 5 delay control. Further, embodiments include multiple delay increments that increase either linearly or nonlinearly as desired. Delay circuits 705 are compact, consume relatively low internal power, and providing high gain. Delay circuits 705 may be used to debug a circuit design, find critical paths, and increase the overall clock speed of an IC by borrowing time from non-critical paths to alleviate a critical path. Once optimal 10 delay settings for application to inputs 720 have been determined, these settings can be fused into the design during mass production/fabrication or during other similar techniques such as bond out or package option.

[0047] FIG. 10 is a block diagram illustrating a demonstrative processing system 1000 for implementing embodiments of the present invention. The illustrated 15 embodiment of processing system 1000 includes one or more processors (or central processing units) 1005, system memory 1010, nonvolatile (“NV”) memory 1015, a data storage unit (“DSU”) 1020, a network link 1025, and a chipset 1030. The illustrated processing system 1000 may represent any computing system including a desktop 20 computer, a notebook computer, a workstation, a handheld computer, a server, a blade server, or the like.

[0048] The elements of processing system 1000 are interconnected as follows. Processor(s) 105 is communicatively coupled to system memory 1010, NV memory 1015, DSU 1020, and network link 1025, via chipset 1030 to send and to receive

instructions or data thereto/therefrom. In one embodiment, NV memory 1015 is a flash memory device. In other embodiments, NV memory 1015 includes any one of read only memory (“ROM”), programmable ROM, erasable programmable ROM, electrically erasable programmable ROM, or the like. In one embodiment, system memory 1010

5 includes random access memory (“RAM”). DSU 1020 represents any storage device for software data, applications, and/or operating systems, but will most typically be a nonvolatile storage device. DSU 1020 may optionally include one or more of an integrated drive electronic (“IDE”) hard disk, an enhanced IDE (“EIDE”) hard disk, a redundant array of independent disks (“RAID”), a small computer system interface 10 (“SCSI”) hard disk, and the like. Although DSU 1020 is illustrated as internal to processing system 1000, DSU 1020 may be externally coupled to processing system 1000. Network link 1025 may couple processing system 1000 to a network such that processing system 1000 may communicate over the network with one or more other computers. Network link 1025 may include a modem, an Ethernet card, Universal Serial 15 Bus (“USB”) port, a wireless network interface card, or the like.

[0049] It should be appreciated that various other elements of processing system 1000 have been excluded from FIG. 10 and this discussion for the purposes of clarity. For example, processing system 1000 may further include a graphics card, additional DSUs, other persistent data storage devices (e.g., tape drive), and the like.

20 Chipset 1030 may also include a system bus and various other data buses for interconnecting subcomponents, such as a memory controller hub and an input/output (“I/O”) controller hub, as well as, include data buses (e.g., peripheral component interconnect bus) for connecting peripheral devices to chipset 1030. Correspondingly,

processing system 1000 may operate without one or more of the elements illustrated.

For example, processing system 1000 need not include network link 1025.

[0050] Delay buffers 705 may be incorporated into processor(s) 1005 or
chipset 1030 to enable the functionality described herein and derive the benefits
5 therefrom. Furthermore, descriptions of IC 700 may be generated, compiled, and/or
tested on processing system 1000. For example, behavioral level code describing IC
700, or portions thereof, may be generated on processing system 1000 using a hardware
descriptive language, such as VHDL or Verilog, and stored to a machine-accessible
medium. Alternatively, processing system 1000 may be used to compile the behavioral
10 level code into register transfer level (“RTL”) code, a netlist, or even a circuit layout of
IC 700. The behavioral level code, the RTL code, the netlist, and the circuit layout all
represent various levels of abstraction to describe IC 700 including delay circuits 705.

[0051] Examples of machine-accessible mediums used to transport the
description of IC 700 include DSU 1020 or other portable media such as a CD-ROM, a
15 DVD, a floppy disk, flash memory, or the like. Alternatively, processing system 1000
may transmit the description of IC 700 out network link 1025 modulated onto a carrier
wave and communicated across a network, such as a local area network, a wide area
network, or the Internet.

[0052] The above description of illustrated embodiments of the invention,
20 including what is described in the Abstract, is not intended to be exhaustive or to limit
the invention to the precise forms disclosed. While specific embodiments of, and
examples for, the invention are described herein for illustrative purposes, various

equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize.

[0053] These modifications can be made to the invention in light of the above detailed description. The terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification and the claims. Rather, the scope of the invention is to be determined entirely by the following claims, which are to be construed in accordance with established doctrines of claim interpretation.